Knowledge Representation And First Order Logic

Saeeda Parsa

sparsa313@gmail.com

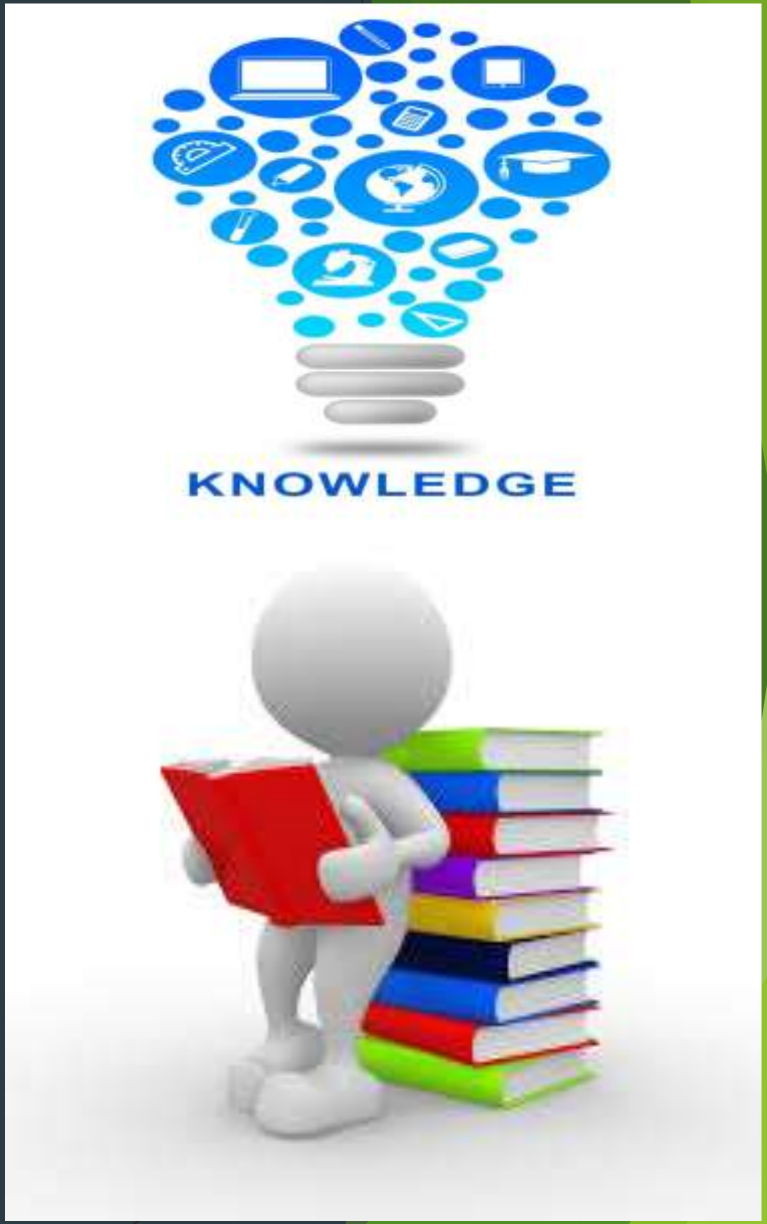
(Sheikh Zayed Islamic Centre,

University of Karachi)

Knowledge Representation:

Knowledge representation is the name we give to how we encode knowledge, beliefs, actions, feelings, goals, desires, preferences, and all other mental states in artificial systems.

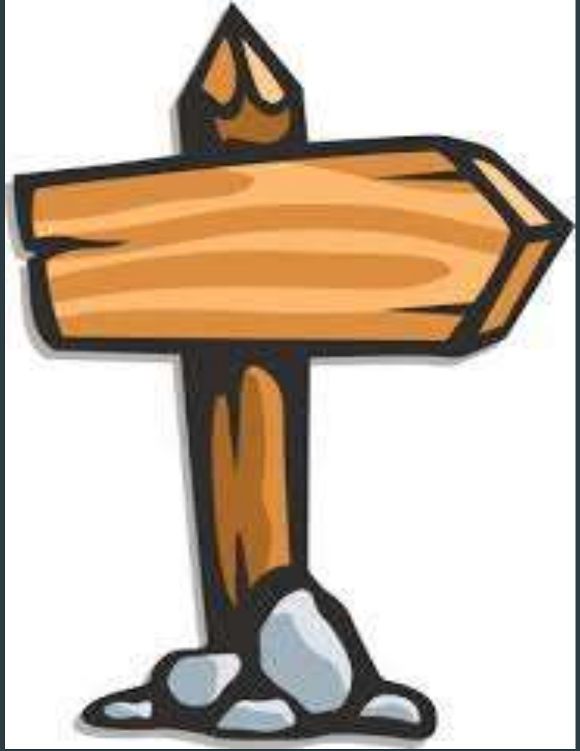
Knowledge is the information about a domain that can be used to solve problems in that domain. To solve many problems requires much knowledge, and this knowledge must be represented in the computer. As part of designing a program to solve problems, we must define how the knowledge will be represented.



sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

What to Represent:

First consider what kind of knowledge which needs to be represented in AI systems:



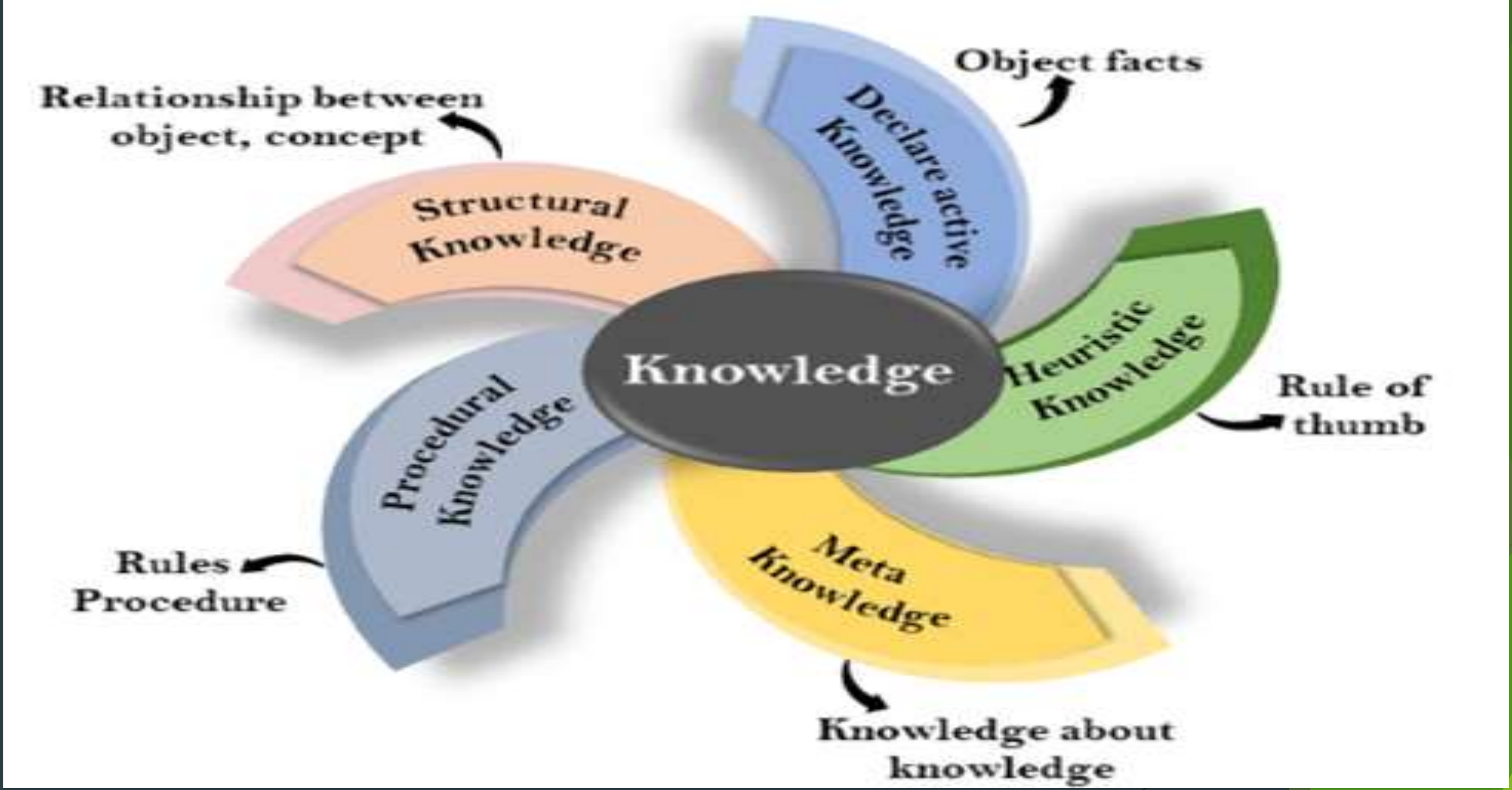
- **Object:** All the facts about objects in our world domain. E.g., Guitars contains strings, trumpets are brass instruments.
- **Events:** Events are the actions which occur in our world.
- **Performance:** It describe behavior which involves knowledge about how to do things.
- **Meta-knowledge:** It is knowledge about what we know.
- **Facts:** Facts are the truths about the real world and what we represent.
- **Knowledge-Base:** The central component of the knowledge-based agents is the knowledge base. It is represented as KB. The Knowledgebase is a group of the Sentences (Here, sentences are used as a technical term and not identical with the English language).

sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

Types of Knowledge in Artificial Intelligence:

1. **Meta Knowledge** – It's a knowledge about knowledge and how to gain them
2. **Heuristic - Knowledge** – Representing knowledge of some expert in a field or subject.
3. **Procedural Knowledge** – Gives information/ knowledge about how to achieve something.
4. **Declarative Knowledge** – Its about statements that describe a particular object and its attributes , including some behavior in relation with it.
5. **Structural Knowledge** – Describes what relationship exists between concepts/ objects.

Types of Knowledge :



sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

General Representations:

- Logical Representations

1. Multi-valued logics
   - More than two truth values
   - e.g., true, false & unknown
   - **Fuzzy logic** uses probabilities, truth value in [0,1]
2. Modal logics
   - Modal operators define mode for propositions
   - **Epistemic logics** (belief)
     - e.g. □p (necessarily p), ◊p (possibly p), …
   - **Temporal logics** (time)
     - e.g. □p (always p), ◊p (eventually p), …

- Non-Logical Representations

1. Production Rules
2. Semantic Networks (Nets)
3. *Description Logics* (Frames)

sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

First-Order Logic

- Definition:
- First-order logic is symbolized reasoning in which each sentence, or statement is broken down into a subject and a predicate. The predicate modifies or defines the properties of the subject. In first order logic a predicate can only refer to a single subject. First order logic is also known as first order predicate calculus or first-order functional calculus.

Syntax

- There are two key parts of first-order logic.
- **Syntax:** determines which collections of symbols are legal expressions in first-order logic, while the semantics determine the meanings behind these expressions.
- **Alphabet:**
- Unlike natural languages, such as English, the language of first-order logic is completely formal so that it can be mechanically determined whether a given expression is legal. There are two key types of legal expressions: which intuitively represent objects and formulas which intuitively express predicates that can be true or false.
- The terms and formulas of first-order logic are strings of symbols where all the symbols together form the alphabet of the language. As with all formal languages the nature of the symbols themselves is outside the scope of formal logic they are often regarded simply as letters and punctuation symbols.

What Is Formal Language?

Formal Language:

Formal specifications use mathematical notation to describe in a precise way the properties which an information system must have, without unduly constraining the way in which these properties are achieved.

A formal specification can serve as a single, reliable reference point for those who investigate the cu

Types of Formal Languages:

1: Z-Language (Z-Notation),

Z is a formal specification language for computer systems which is based on set theory and predicate logic.

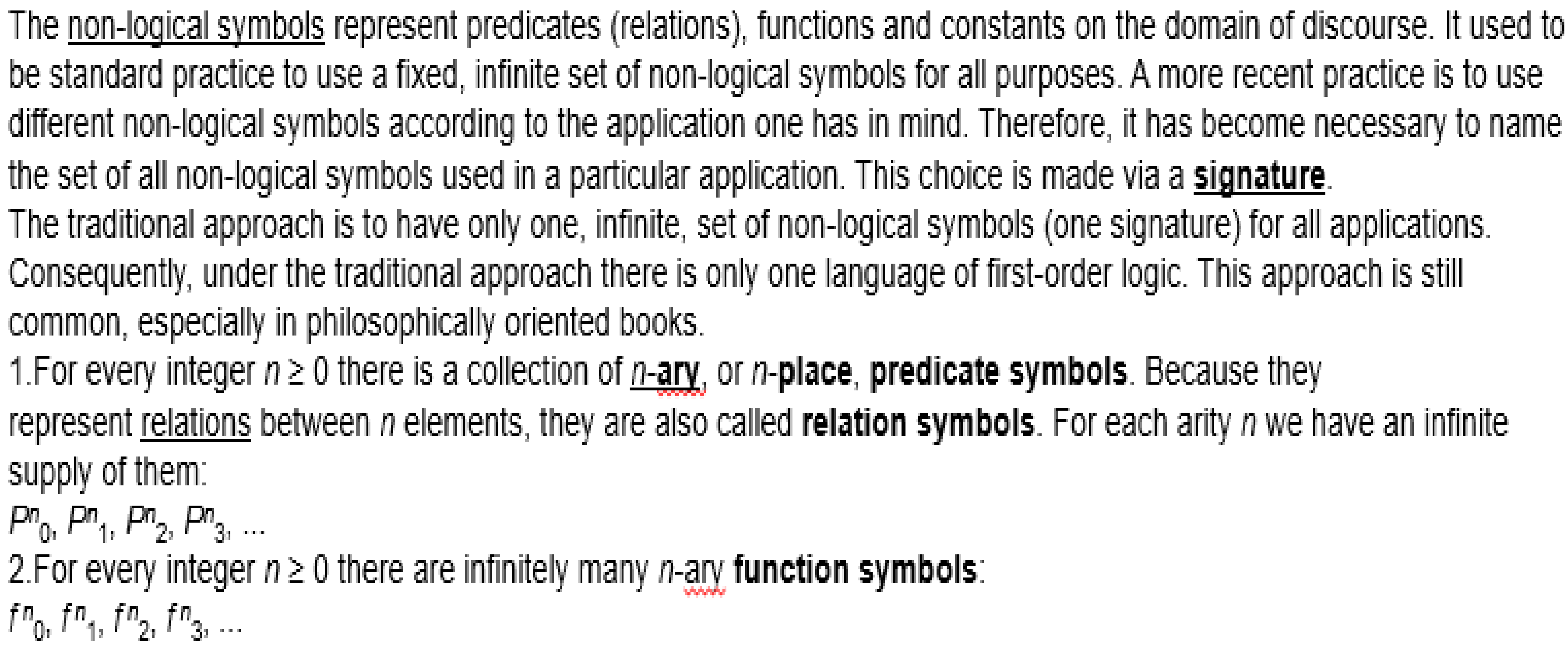
2: Prolog (Programming Language)

sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

Logical Symbols:

- There are several logical symbols in the alphabet, which vary by author but usually include:
- The quantifier symbols ∀ and ∃
- The logical connectives: ∧ for conjunction, ∨ for disjunction, → for implication, ↔ for biconditional, ¬ for negation. Other logical connective symbols are included. Some authors use C*pq*, instead of →, and E*pq*, instead of ↔, especially in contexts where → is used for other purposes. Moreover, the horseshoe ⊃ may replace →; the triple-bar ≡ may replace ↔.
- Parentheses, brackets, and other punctuation symbols.
- An infinite set of variables. Likely alphabet x, y, z, … . Subscripts are often used to distinguish variables: x_0, x_1, x_2, … .
- An equality symbol (sometimes, identity symbol) =; see the section on equality below.

Non-logical symbols



The non-logical symbols represent predicates (relations), functions and constants on the domain of discourse. It used to be standard practice to use a fixed, infinite set of non-logical symbols for all purposes. A more recent practice is to use different non-logical symbols according to the application one has in mind. Therefore, it has become necessary to name the set of all non-logical symbols used in a particular application. This choice is made via a **signature**.

The traditional approach is to have only one, infinite, set of non-logical symbols (one signature) for all applications. Consequently, under the traditional approach there is only one language of first-order logic. This approach is still common, especially in philosophically oriented books.

1.For every integer $n \geq 0$ there is a collection of *n*-**ary**, or *n*-**place**, **predicate symbols**. Because they represent relations between n elements, they are also called **relation symbols**. For each arity n we have an infinite supply of them:

$P^n_0, P^n_1, P^n_2, P^n_3, \ldots$

2.For every integer $n \geq 0$ there are infinitely many *n*-ary **function symbols**:

$f^n_0, f^n_1, f^n_2, f^n_3, \ldots$

sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

Formulas:

1. Predicate symbols. If P is an n-ary predicate symbol and t1, ..., tn are terms then $P(T_1....T_n)$ is a formula.

2. Equality. If the equality symbol is considered part of logic, and tl and t2 are terms, then t1 = t2 is a formula.

3. Negation. If cp is a formula, then —Q is a formula.

4. Binary connectives. If cp and are formulas, then (Q * W) is a formula. Similar rules apply to other binary logical connectives.

5. Quantifiers. If p is a formula and x is a variable, then (for all x, p holds) and (there exists x such that p) are formulas.

sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

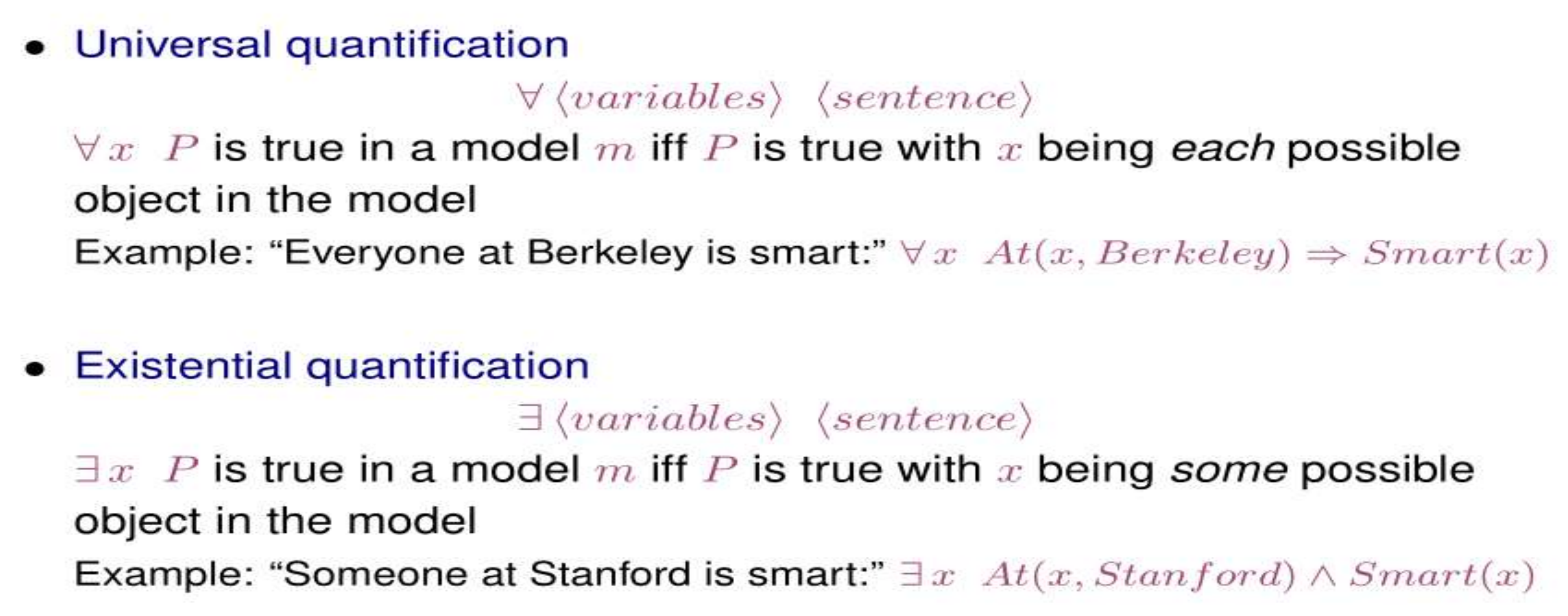
Quantifiers

A language element which generates a quantification (such as "every") is called a quantifier.

First-order logic contains two standard quantifiers, called universal and existential.

sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

Types Of Quantifiers:



- Universal quantification

$$\forall \langle variables \rangle \quad \langle sentence \rangle$$

$\forall x \;\; P$ is true in a model m iff P is true with x being *each* possible object in the model

Example: "Everyone at Berkeley is smart:" $\forall x \;\; At(x, Berkeley) \Rightarrow Smart(x)$

- Existential quantification

$$\exists \langle variables \rangle \quad \langle sentence \rangle$$

$\exists x \;\; P$ is true in a model m iff P is true with x being *some* possible object in the model

Example: "Someone at Stanford is smart:" $\exists x \;\; At(x, Stanford) \wedge Smart(x)$

sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

Predicate

- in mathematical logic a predicate is commonly understood to be a Boolean-valued function $P: X \to \{\text{true}, \text{false}\}$, called the predicate on X. However, predicates have many different uses and interpretations in mathematics and logic, and their precise definition, meaning and use will vary from theory to theory. So, for example, when a theory defines the concept of a relation then a predicate is simply the characteristic function(otherwise known as the indicator function) of a relation. However, not all theories have relations, or are founded on set theory and so one must be careful with the proper definition and semantic interpretation of a predicate.

Semantics:

- An interpretation of a first-order language assigns a denotation to each non-logical symbol in that language. It also determines a domain of discourse that specifies the range of the quantifiers. The result is that each term is assigned an object that it represents, each predicate is assigned a property of objects, and each sentence is assigned a truth value. In this way, an interpretation provides semantic meaning to the terms, the predicates, and formulas of the language. The study of the interpretations of formal languages is called formal semantics. What follows is a description of the standard or Tarskian semantics for first-order logic. (It is also possible to define game semantics for first-order logic but aside from requiring the axiom of choice, game semantics agree with Tarskian semantics for first-order logic, so game semantics will not be elaborated herein.)

Uses And History:

- First-order logic can be useful in the creation of computer programs. It is also of interest to researchers in artificial intelligence (AI). There are more powerful forms of logic, but first-order logic is adequate for most everyday reasoning. The Incompleteness Theorem , proven in 1930, demonstrates that first-order logic is in general undecidable. That means there exist statements in this logic form that, under certain conditions, cannot be proven either true or false.

Logic And Mathematics

- In mathematics, it is used in an informal way quite extensively. The various connectives and quantifiers find their way into a great deal of mathematical discourse, both formal and informal, as in a classroom setting. Here you will often find elements of FOL interspersed with English or the mathematician's native language. If you've ever taken calculus you have probably seen such formulas as:
- $\forall \epsilon > 0 \; \exists y > 0 \ldots$
- Here, the unusual, rotated letters are taken directly from the language FOL.

sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

Logic And Philosophy

- In philosophy, FOL and enrichments of it are used in two different ways. As in mathematics, the notation of FOL is used when absolute clarity, rigor, and lack of ambiguity are essential. But it is also used as a case study of making informal notions (like grammaticality, meaning, truth, and proof) precise and rigorous. The applications in linguistics stem from this use, since linguistics is concerned, in large part, with understanding some of these same informal notions.

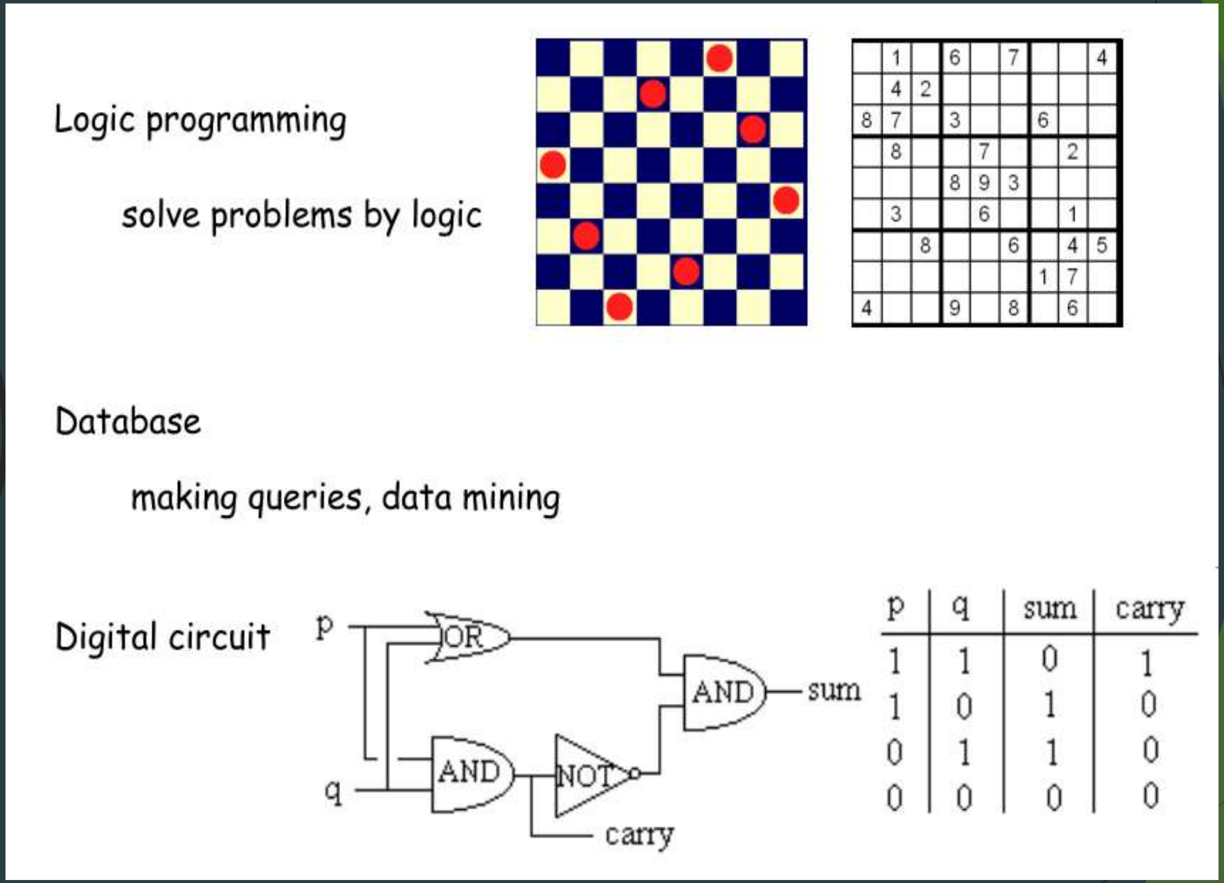
Logic And Artificial Intelligence

- In artificial intelligence, FOL is also used in two ways. Some researchers take advantage of the simple structure of FOL sentences to use it as a way to encode knowledge to be stored and used by a computer. Thinking is modeled by manipulations involving sentences of FOL. The other use is as a precise specification language for stating axioms and proving results about artificial agents.

Logic And Computer Science

- In computer science, FOL has had an even more profound influence. The very idea of an artificial language that is precise yet rich enough to program computers was inspired by this language. In addition, all extant programming languages borrow some notions from one or another dialect of FOL.

Applications of Logic



Logic programming

solve problems by logic

	1		6		7			4
	4	2						
8	7		3			6		
	8			7			2	
			8	9	3			
	3			6			1	
		8			6		4	5
						1	7	
4			9		8		6	

Database

making queries, data mining

Digital circuit

p
OR
AND
sum
q
AND
NOT
carry

p	q	sum	carry
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

sparsa313@gmail.com (Sheikh Zayed Islamic Centre)

Thank You

sparsa313@gmail.com (Sheikh Zayed Islamic Centre)